Introductory Course: Using LS-OPT® on the TRACC Cluster

1.5 - Using LS-OPT on the TRACC High Performance Cluster

By: Cezary Bojanowski, PhD

Generating LS-OPT Job Scripts

- Navigate to the folder with the LS-OPT input file `<your-opt-input-file>`

- Where: `<your-opt-input-file>` is typically a `com.file_name` file

- The same command that was submitting LS-DYNA jobs to PBS `qsub-mpp-dyna` will run LS-OPT jobs when used with the option: `--lsopt <your-opt-input-file>`

- For example: `qsub-mpp-dyna --lsopt <your-opt-input-file> …`

- This operation will cause `lsoptscript` and `job script` to be generated and automatically submitted to PBS.

- Note: `lsoptscript` with full path is your solver in `com.file_name`

Running LS-OPT in Graphical Mode

- To run LS-OPT in GUI generate the scripts without submitting them to PBS. Use `--nosubmit` option with the previous command.

- For example: `qsub-mpp-dyna --quad --lsopt <your-opt-input-file> --cores 32 --hours 10 –N <name-of-job-script> --nosubmit`

- Afterwards run NoMachine NX application and start LS-OPT GUI with `lsoptui` command.

- Load `<your-opt-input-file>` to the program.

- If you choose to load one of LSTC examples, the file with `com.name.correct` in its name (For example: `com.linear.correct`) should be almost ready to be run.

Example

- Only `--lsopt --nosubmit` are added to a normal LS-DYNA command

Example

pbsjob script


```
#!/bin/sh

## This file was created at Mon Dec  7 00:17:10 CST 2009 with the
command
##        qsub-mpp-dyna-beta --lsopt com.PICKUP-FRONT-IMPACT-REDUCED --
cores 4 memory1=200m memory2=70m --module pgi-hp/r4.2.1 --hours 6 --
nosubmit

## To run this job, do
##        cd /home/cbojanowski/scratch/PICKUP-FRONT/TEST; lsopt
com.PICKUP-FRONT-IMPACT-REDUCED

#PBS -j oe
#PBS -l walltime=6:00:00
#PBS -l select=1
#PBS -l mppdyna=4

export LSTC_BINARY=32ieee

. /etc/profile.d/modules.sh
module unload ls-dyna/mpp ls-dyna/smp
module load ls-dyna/base ls-dyna/mpp/pgi-hp/r4.2.1

cd $PBS_O_WORKDIR

exe=`which mpp_d`
mpicmd="wrapper mpirun -hostfile $PBS_NODEFILE $exe   i=DynaOpt.inp
memory1=200m memory2=70m"

echo 'Starting job at' `date`
echo 'Executing command:'
echo $mpicmd

$mpicmd

echo 'Ending job at' `date`

trap 'onexit' EXIT TERM
function onexit {
        if [ "$already" != "true" ]; then
                already=true
                awk '{print "This job was (PID@node) " $5 "@" $2 "."}'
bg_switch
```

clean script


```
#!/bin/sh

## clean
## removes superfluous LS-OPT data
## see LS-OPT manual, section 9.9

rm -rf d3* elout nodout rcforc
```

lsoptscript


```
qsub -v LSOPT_PORT,LSOPT_HOST,
../../pbsjob.14267
```

Running LS-OPT in Graphical Mode

- Specify Solver Command on Solvers page

`/full/absolute/path/to/lsoptscript`
`/mnt/gpfs/home/cbojanowski/scratch/DUMMY/OPT/lsoptscript`
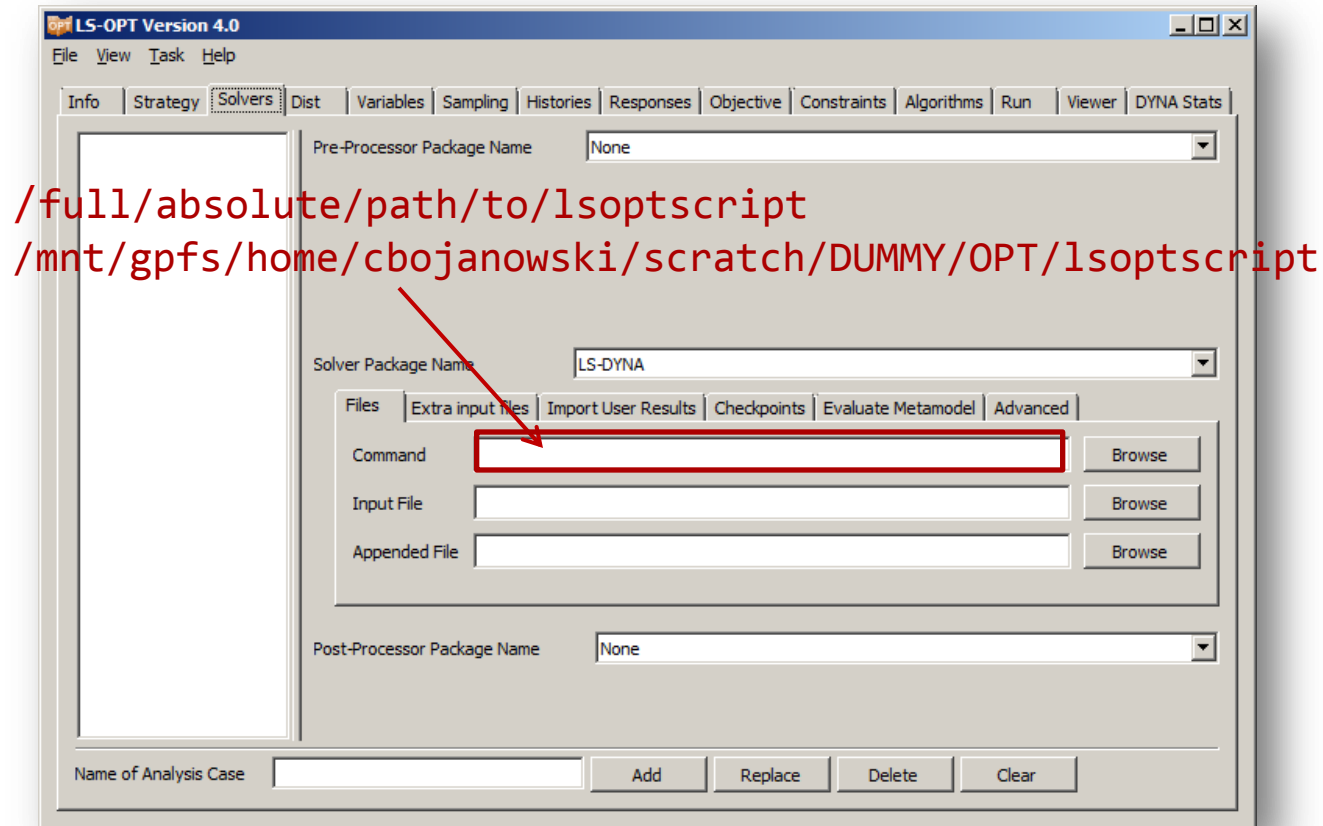
Running LS-OPT in Graphical Mode

- Set PBS as a job scheduler on Run page
- Specify number of concurrent jobs – <u>it is better to submit more small jobs</u>
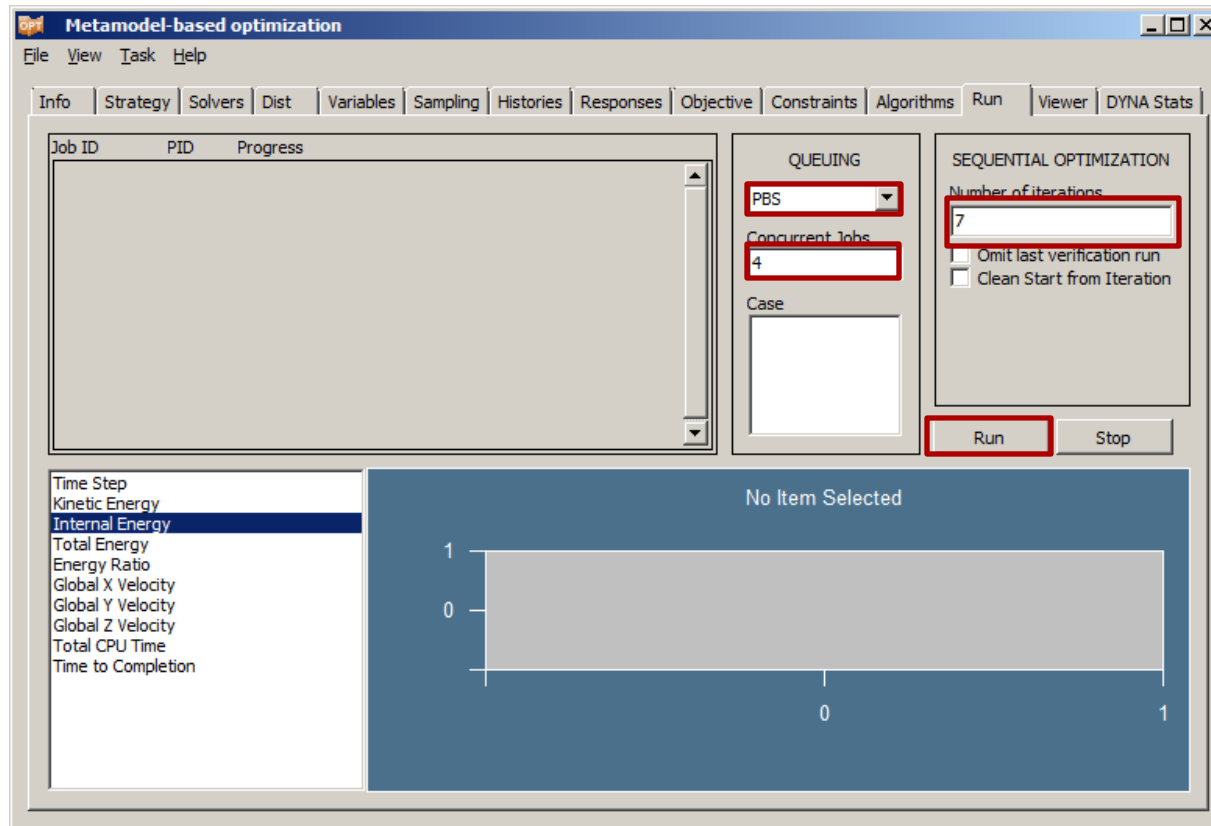
Clean Script

- With the `job script` and `lsoptscript` also `clean` script is generated. It contains the syntax:

- `rm -rf d3* elout nodout rcforc`

- To reduce the disk space required by multiple runs this script automatically deletes `d3plot, d3dump, elout, nodout` and `rcforc` files from all folders but `1.1` and the `<last-optimal-run>.1`

- If required, you can request disabling the clean file with `--noclean` option when running `qsub-mpp-dyna` command or modify existing one to furnish your needs.

Work Directory Structure

Command file, input files
Output files

Database files

Simulation files
Intermediate files
Status files,
plot files

Work directory

Case 1   Case 2

1.1   1.2   ...   m.n   1.1   1.2   ...   m.n
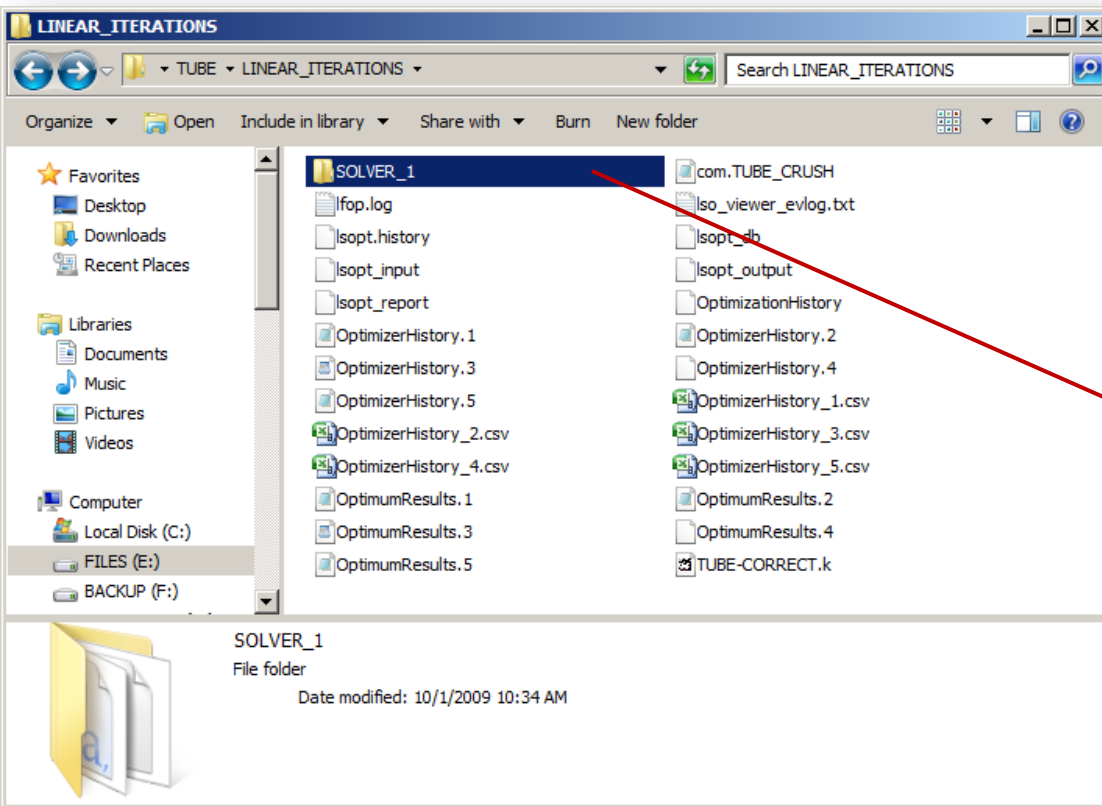
Run directories
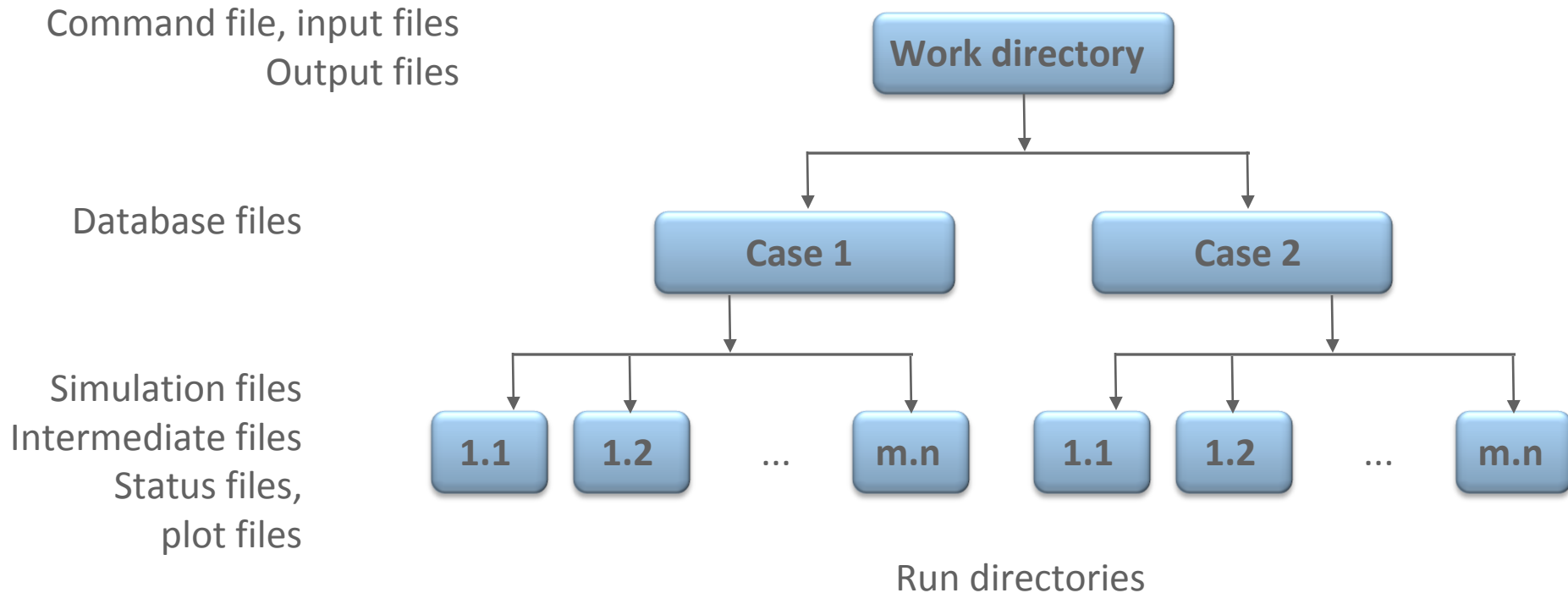
n – number of design points (simulations per iteration)
m – number of iterations

Work Directory Structure

work directory

case directory
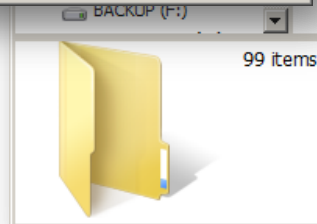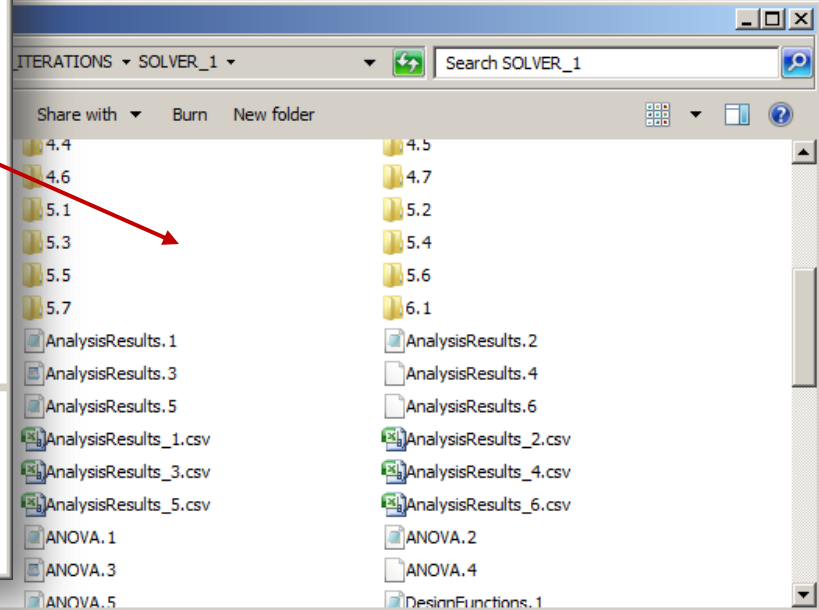
ASCII Database Files

Database file	Description	Location
Experiments	Trial designs computed as a result of the experimental design	Case
AnalysisResults	Trial designs and the responses extracted from the solver database	Case
DesignFunctions	Parameters of the approximate functions	Case
OptimizationHistory	Variable, response and error history of approximation process	Case
OptimizerHistory	Detailed history of optimizer	Work
TradeOff	All variable, responses and results of the non-dominated solutions at each iteration	Work
ExtendedResults	All variable, responses and results at each trial design point	Case
Net.funcname	Parameters of the metamodel for function funcname	Case

Output files

Database file	Description	Location
lsopt_input	input in a formatted style	Work
lsopt_output	results and some logging information	Work
lsopt_report	a final report of the analysis results available for some main and the Repair tasks	Work
history_design	Table of the objective and constraint values for each iteration	Work
history_variables	Table of the design variables, responses and composites values for each iteration	Work
lsopt_db	status of LSOPT for other LSTC programs	Work

Results in .csv Format

Database file	Description	Location
Experiments_n.csv	Experiments (n – iteration number)	Case
AnalysisResults_n.csv	Analysis results	Case
ExtendedResultsMaster_n.csv	Extended Results for variables, responses, composites, objectives and constraints	Case
ExtendedResultsMETAMaster_n.csv	Extended Results for user defined Experiments	Case
PRESS_predictions_n.csv	PRESS residuals and predicted results	Case
OptimizarHistory_n.csv	Detailed history of the optimizer for iteration n	Work